

AMENDMENTS TO THE CLAIMS

Please add new claims 18-48 as follows:

18. (New) A method of processing a data stream in a general purpose processor capable of operation independent of another host processor, the general purpose processor having a virtual memory addressing unit, an instruction path and a data path to digitally process the data stream, the method comprising:

receiving the data stream over the data path;

dynamically partitioning the data stream based on an elemental width of the data and storing partitioned data in registers of a register file coupled to the data path, wherein a number of data elements stored in a register is inversely related to the elemental width of the data stored in partitioned fields of the register;

performing group floating point operations on multiple operands stored in partitioned fields of registers and, for each group floating point operation, returning catenated results of the operation to a register.

19. (New) The method of claim 18 wherein the data stream comprises media data.

20. (New) The method of claim 19 wherein the media data comprises broadband communications data.

21. (New) The method of claim 19 wherein the media data comprises audio data.

22. (New) The method of claim 19 wherein the media data comprises image data.

23. (New) The method of claim 19 wherein the media data comprises video data.

24. (New) The method of claim 19 wherein the media data comprises compressed data.

25. (New) The method of claim 19 wherein the media data comprises error checking data.
26. (New) The method of claim 19 wherein the media data comprises error correction data.
27. (New) The method of claim 18 wherein, for a specific group floating point operation performed, the catenated results of the specific operation are returned to a register that is different than the registers used to store the multiple operands for the specific operation.
28. (New) The method of claim 18 wherein the performing step comprises performing group add, group subtract and group multiply arithmetic operations on catenated floating-point data and, for each such group operation, returning catenated results of the operation to a register.
29. (New) The method of claim 18 wherein the performing group floating-point operations comprises operating, in parallel, on multiple operands stored in partitioned fields of registers.
30. (New) The method of claim 18 wherein the performing group floating-point operations comprises performing a first group floating-point operation on floating-point data of a first precision and performing a second group floating-point operation on floating-point data of a second precision that is a lower precision than the first precision by operating, in parallel, on at least two operands stored in partitioned fields of registers.
31. (New) The method of claim 18 further comprising performing group integer operations on multiple operands stored in partitioned fields of registers and, for each group integer operation, returning catenated results of the operation to a register.

32. (New) The method of claim 31 wherein, for a specific group integer operation performed, the catenated results of the specific operation are returned to a register that is different than the registers used to store the multiple operands for the specific operation.

33. (New) The method of claim 31 wherein the performing group integer operations comprises performing group add, group subtract and group multiply arithmetic operations on catenated integer data and, for each such group operation, returning catenated results of the operation to a register.

34. (New) The method of claim 31 wherein the performing group integer operations comprises operating, in parallel, on multiple operands stored in partitioned fields of registers.

35. (New) The method of claim 31 wherein the performing group integer operations comprises performing a first group integer operation on integer data of a first precision and performing a second group integer operation on integer data of a second precision that is a lower precision than the first precision by operating, in parallel, on at least two operands stored in partitioned fields of registers.

36. (New) The method of claim 18 further comprising performing one or more group data handling operations that operate on multiple operands stored in partitioned fields of operand registers and, for each group data handling operation, returning catenated results of the operation to a register.

37. (New) The method of claim 36 wherein the performing one or more group data handling operations comprises converting a plurality of n -bit data elements in a first operand register and a plurality of n -bit data elements in a second operand register into a plurality of $n/2$ -bit data elements.

38. (New) The media processor of claim 37 wherein the converting step shifts each of the plurality of $n/2$ -bit data elements by a specified number of bits during the conversion.

39. (New) The method of claim 36 wherein the performing one or more group data handling operations comprises interleaving a plurality of data elements selected from a first operand register with a plurality of data elements selected from a second operand register and concatenating the data elements into a result register.

40. (New) The method of claim 36 wherein the performing one or more data handling operations comprises shifting bits of individual data elements concatenated in an operand register to the left and clearing empty low order bits of the individual data elements to zero.

41. (New) The method of claim 36 wherein the performing one or more data handling operations comprises shifting bits of individual data elements concatenated in an operand register to the right and filling empty high order bits of the individual data elements with a value equal to a value stored in a sign bit of the individual data element.

42. (New) The method of claim 36 wherein the performing one or more data handling operations comprises shifting bits of individual data elements concatenated in an operand register to the right and clearing empty high order bits of the individual data elements to zero.

43. (New) The method of claim 36 wherein the performing one or more group data handling operations comprises operating, in parallel, on multiple operands stored in partitioned fields of registers.

44. (New) The method of claim 36 wherein the performing one or more group data handling operations comprises performing a first group data handling operation on data of a first precision and performing a second group data handling operation on data of a second precision

that is a lower precision than the first precision by operating, in parallel, on at least two operands stored in partitioned fields of registers.

45. (New) The method of claim 18 wherein the group floating point operations are associated with a plurality of instruction streams from a plurality of threads executing in parallel on the processor.

46. (New) The method of claim 18 wherein the performing step comprises performing group floating-point operations on data having a total aggregate width of 128 bits.

47. (New) The method of claim 18 wherein the performing step comprises performing group floating-point operations on data of more than one precision.

48. (New) The method of claim 18 further comprising storing floating-point data in a register file in a format conforming to IEEE standard 754.
